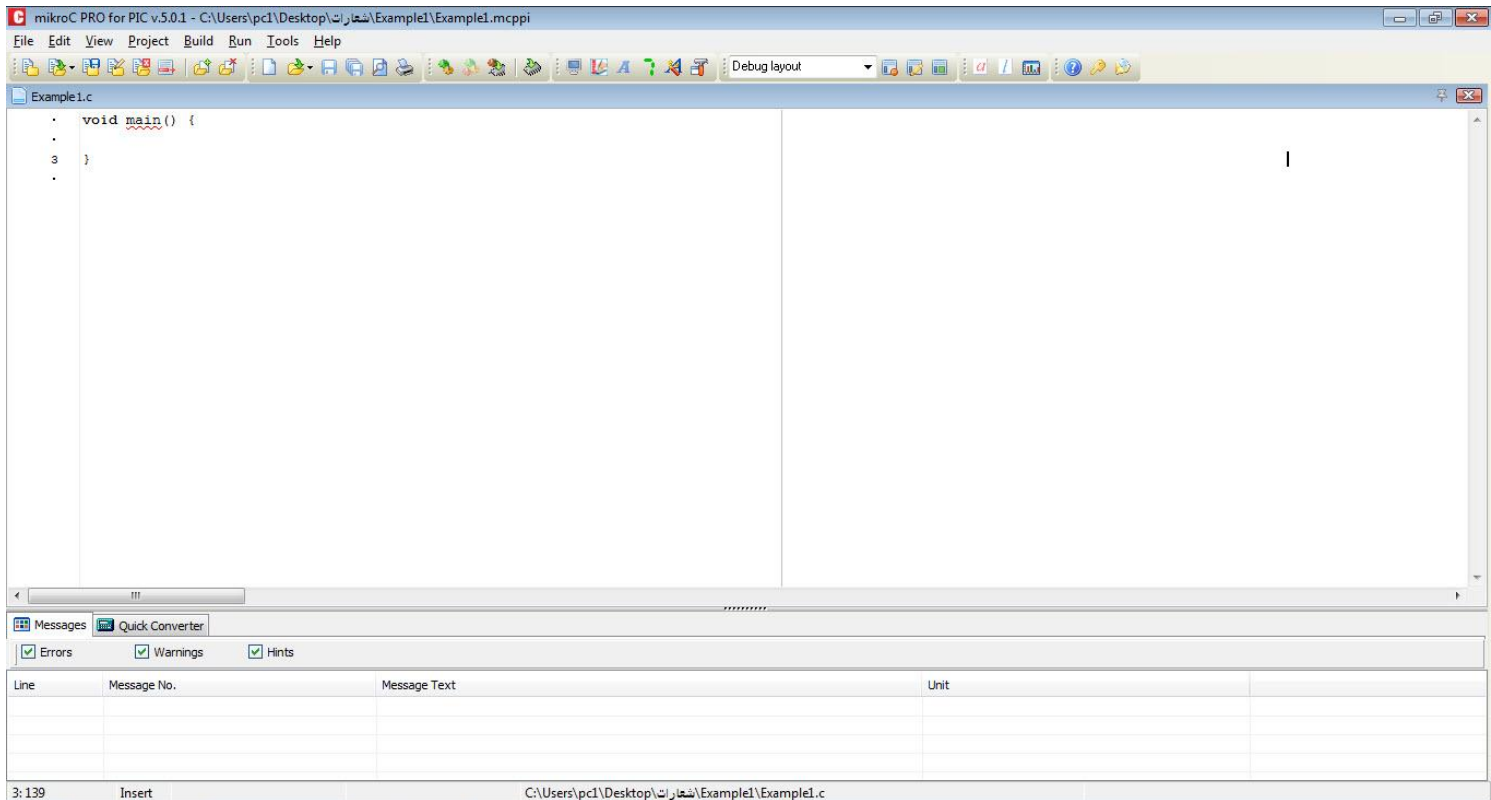


ذكرنا في المحاضرة السابقة اننا سنستخدم برنامج المايكروسي الذي يقوم بترجمة البرنامج سطر سطر وتنفيذه بشكل متسلسل ولتتمكن من زيادة قوتك في البرمجة لابد لك من التعامل مع عدد كبير من الامثلة وخاصة في البداية وهذا ما سنفعله في هذه الدورة .

والآن افتح برنامج المايكروسي ونفذ مشروعاً جديداً وليكن اسمه Example1 واختر المتذبذب 4MHz واختر المتحكم 16f877a سنصل الى هذه نافذه



اذا فان البرنامج يكتب في هذا المكان بين القوسين

```
Void main() {
    هنا نكتب الكود الخاص بالبرنامج
}
```

والان سنتطرق الى عدد من أوامر المايكروسي وأهمها من خلال البرنامج البسيط التالي

```
Void main() {
    Trisb=0b00000000;
    Portb=0b00001111;
}
```

لنشرح هذا الكود السطر الاول يكتب في بداية أي برنامج
السطر الثاني يوظف جميع الارجل من B7-B0 كخرج حيث صفر يعن خرج و واحد يعني دخل
السطر الثالث يقوم بأخراج خمسة فولت على الارجل من B7-B4 والبقية تكون صفر فولت
والسطر الرابع هو اغلاق القوس

خلاصه : عزيزي القارئ ان البيك لا يستطيع ان يخرج فولت على احد اطرافه الان اذا اخبرته ان هذا الطرف هو خرج
ولكن كيف سأخبره؟؟؟؟ يتم ذلك من خلال الامر TRIS فمثلاً لو اردت ان اجعل اي طرف من اطراف PORTB خرج
نكتب TRISB ولكن ما هي القيمة التي سنخزنها في الرجستر TRISB بسيط جداً

ان هذا الرجيستر هو عبارته عن ٨ بت رجستر ماذا يعني ٨ بت ؟؟ ... أي ان القيمة المخزنة به لو كتبناها بالبايناري
ستحتوي على ثماني خانات كل بت(خانه) تعبر عن رجل من أجل PORTB وبهذا لو اردنا ان نجعل B0 و B1 فقط هي
خرج عندها سنكتب الجملة التالية بحيث اول خانتين فقط هما صفر (خرج) والبقية واحد (دخل).

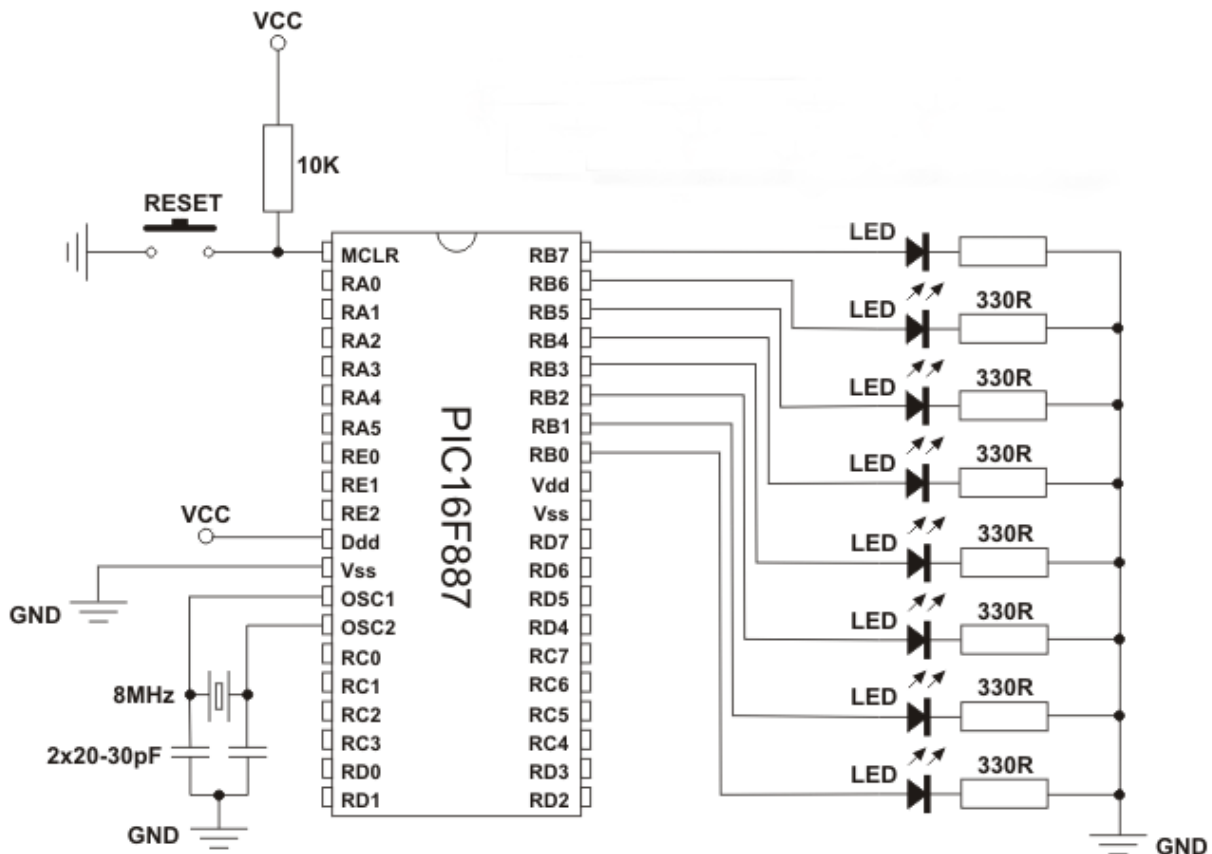
```
Void main() {
    Trisb=0b00111111;
}
```

ملاحظة : كل ما ينطبق على PORTB ينطبق على جميع الـ PORTs دون أي استثناء (سوا حالات بسيطة جداً)

والان لنعد الى برنامجنا البسيط الاول وهو

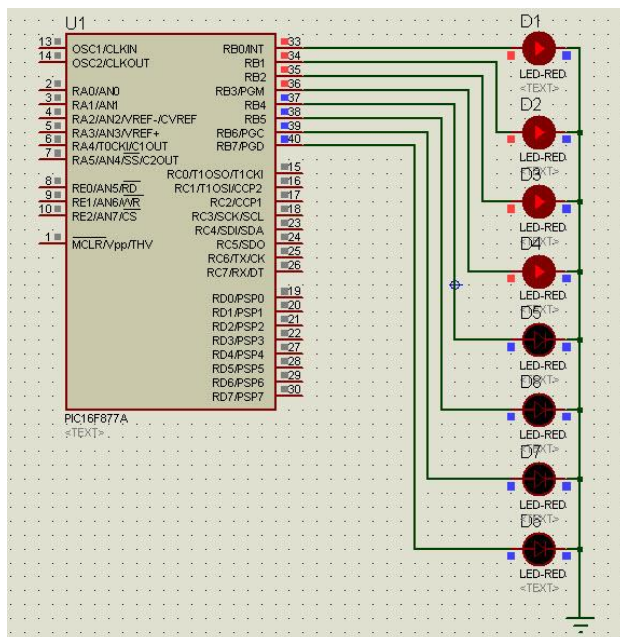
```
Void main() {
    Trisb=0b00000000;
    Portb=0b00001111;
}
```

لوقما بتوصيل ليد (LED) على اطراف الـ PORTB شرط ان يكون على B7,B6,B5,B4 ما تتوقع ان يحدث؟؟
يؤدي هذا الى اضاءة اليدات الموصلة ولمعرفة طريقة التوصيل انظر الى الصورة التالية



للاحظ في الصورة الموجوده في الاعلى اننا وصلنا كل التوصيلات الاساسية للمايكروكنترولر.... وايضا قمنا بتوصيل اليدات على كل PORTB فماذا نتوقع ان يحدث؟

قم بانشاء البرنامج على المايكروسي وارسم الدائره على برنامج بروتس وادخل البرنامج على البيك من خلال ملف الـ HEX كما تم شرحه من خلا المحاضره السابقه وشاهد كا النتيجة وبعد القيام بذلك ستكون النتيجة هي كالتالي :



لكن هل تلاحظ شيء غريباً؟؟

لاحظ اننا لم نوصل التوصيلات الاساسية في المايكروكنترولر على ملف المحاكاة.... نعم اخي القارئ ان برنامج بروتس يفترض ان هذه التوصيلات هي توصيلات معروفه لديه وروتينه فلا داعي من توصيلها فهي معرفه لديه ضمناً .

والان لقد تعلمنا كيف يمكننا ان كتابة برنامج بسيط على المايكروسي وتنفيذه على برنامج البروتس لذلك في المرات القادمة سيكون الشرح اقل تفصيلاً وخاصة بالنسبة للبروتس لذلك الان سندخل في بعض قوعد البرمجة بلغة المايكروسي

الجملة الشرطية IF

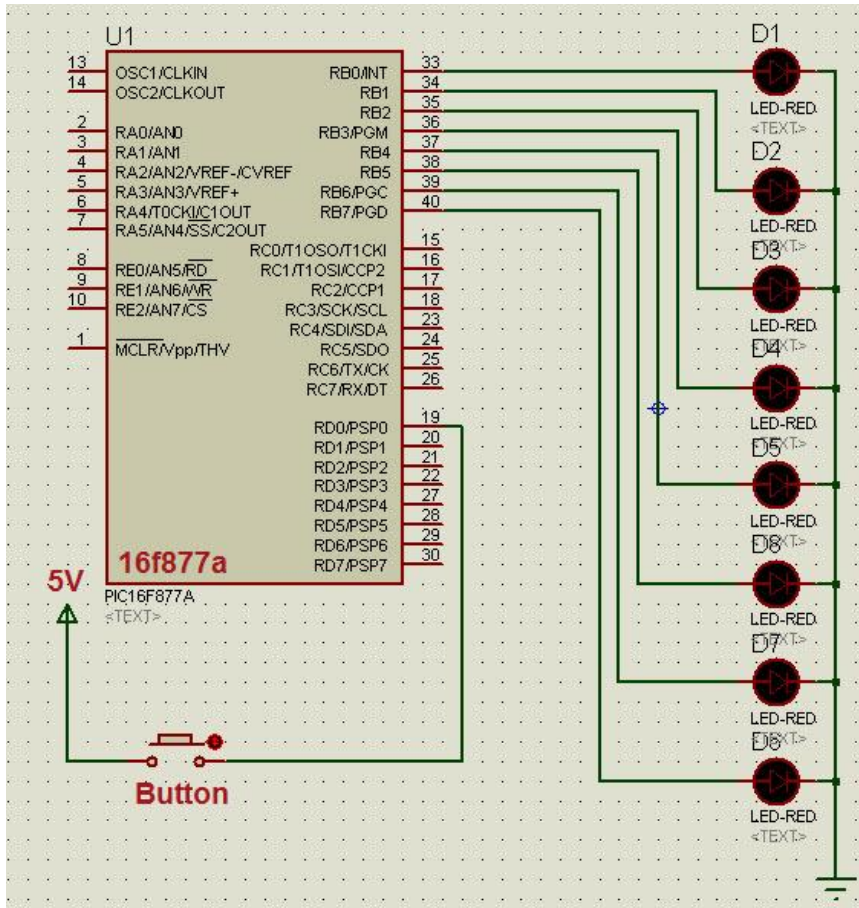
من المعروف ان المبدأ الاساسي لجملة الشرطية هو تنفيذ أمر معين في حلة تحقيق شرط ما ... وهذه هي الصيغة العالمه لجملة الشرطه في برنامج المايكروسي

```
{الاوامر التي تنفذ في حالة تحقق الشرط} (الشرط الواجب تحقيقه) IF
Else
{الاوامر المراد تحقيقها في حالة عدم تحقق الشرط}
```

وهنا لابد من التنويه في بعض الامور منها انه من الممكن ان نكتب كل الجملة على سطر واحد ولكن بتأكيد هذا يجعل البرنامج ذر شكل غير لائق كما انه من الممكن ان نستخدم المجله الشرطية دون استخدام Else أي انه اذا تحقق الشرط نفذ الاوامر وفي حال عدم تحقق الشرط لا تنفذ شيء

ما رأيكم ان نأخذ مثلاً بسيطاً على الجملة الشرطية فمثلاً قم برسم الدائرة التالية على برنامج بروتس ونفذ البرنامج التالي على برنامج المايكروسي ولاحظ عندما نضغط على الـ Button ما يحدث

```
void main () {
    trisb=0x00;
    trisd=0xff;
loop:
    {if (portd.f0==0) {portb=0x00; }
    else
    {portb=0xff; }
    }goto loop
}
```



لاحظ اخي القارئ اننا قمنا بتوصيل الـ Button على PORTD وعلى الرجل رقم صفر منه هذا يعني ان هذه الرجل هي دخل لانها ستتقبل ه فولت عند الضغط على الـ Button وايضا PORTB هو مخرج لانه موصل مع الـ LEDs

والان استخدمنا جملة شرطية لنسأل هل تم الضغط على المفتاح؟ أي هل اصبحه الرجل الاولى من PORTD بـ Logic 1 (ه فولت) فإذا تنفذ الشرط اذاً قم بتنفيذ الامر الذي هو تشغيل الـ LEDs الموصلة على PORTB

ولعل البعض منكم يتساءل عن الجملتين

LOOP و goto loop وهذه الجملتين تفيد في عملية التكرار لانه يطلب على المايكروكنترولر اعادة السؤال بشكل دائم والا سيأخذ القيمة الاولى ويتوقف لتفهم ذلك بشكل أوضح قم بمسح هاتين الجملتين ونفذ البرنامج ولاحظ ماذا يحدث

جملة التكرار FOR

لاحظت عزيزي القارئ في المثال السابق اهمية جملة LOOP التي تفيد التكرار بشكل لا نهائي ولكن ماذا لو اردنا التكرار لعدد معين من المرات بتأكيد نحن بحاجة الى جملة FOR وما يلي هي الصيغة العامة لجملة

```
void main () {
    ( مقدار الزيادة ; الشرط ; القيمة الابتدائية )
    for
    {
        الاوامر المراد تكرارها
    }
}
```

وهنا لنأخذ مثال على ذلك كما في الكود التالي :

```
void main () {
    trisb=0x00;

    int x ;

    for (X=1;x=10;x++)
    {
        Portb=0x00;
        Delay_ms (2000) ;
        Portb=0xff;
        Delay_ms (2000) ;
    }
}
```

لاحظ هنا انه يجب في البداية ان نعرف المتغير X على انه int ومن ثم ندخل الى جملة التكرار التي تفيد تكرار الاوامر ١٠ مرات فقط ولاحظ ايضاً اننا استخدمنا جملة جديده وهي Delay_ms وهي تفيد في عمل توقف للبرنامج لمدته معينه وذلك حسب الرقم الموجود داخلها والرقم الذي بداخلها هو قيمت الوقت ويكتب بلملي ثانية كما ويمكن استخدام الجملة التالية Delay_us التي يكتب بداخلها الوقت بوحدة المايكروثانية .

قم بتطبيق البرنامج على نفس الدائرة السابقة ولحظ النتيجة

كما ويمكن استخدام جملة for لتفيد التكرار بشكل لانهاهي وذلك من خلال الصيغة التالية

```
void main () {
    for(;;)
    {
        الاوامر المراد تكرارها
    }
}
```

قم بتطبيق باستخدام جملة for لتفيد التكرار الانهائي في المثال الذي استخدمنا فيه الجملة الشرطة

والى هنا نكون قد وصلنا الى نهاية المحاضرة الثانية وننتظر منك الملاحظات

وفي المحاضرة الثالثة سيكون هناك عدد من الامثلة التطبيقية للوصول الى مستوى جيد من التعامل مع لغة المايكروسي
وبرنامج البروتس